Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	38	(Kolagotla near2 Ravi).in.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/10 16:58
L2	44	(Juan near2 Revilla).in.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/10 16:58
L3	3	1 and 2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/10 16:58
L4	85420	cache	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/10 16:58
L5	38285	USB	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/10 16:58
L6	724	L4 same L5	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/10 16:58
L7	50449	SRAM	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/10 16:58
L8	112	L6 AND L7	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/10 16:58
L9	130047	hit or miss	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/10 16:58

L10	1448331	index or descriptor$2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/10 16:58
L11	20975	L9 and L10	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/10 16:58
L12	20	L11 and L8	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/10 16:58
L13	94966	hierarchy or hierarchical	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/10 16:58
L14	1	L12 and L13	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/07/10 17:00

PORTAL USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: The ACM Digital Library The Guide

+page +descriptor, +cache +index, +TLB, +mapping, +hierarc

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used
page descriptor cache index TLB mapping hierarchical

Found **32** of **157,873**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 32 Result page: **1** 2 next

Relevance scale

1 Page placement algorithms for large real-indexed caches

R. E. Kessler, Mark D. Hill

November 1992 **ACM Transactions on Computer Systems (TOCS)**, Volume 10 Issue 4

Full text available: pdf(1.55 MB) Additional Information: full citation, abstract, references, citings, index terms

When a computer system supports both paged virtual memory and large real-indexed caches, cache performance depends in part on the main memory page placement. To date, most operating systems place pages by selecting an arbitrary page frame from a pool of page frames that have been made available by the page replacement algorithm. We give a simple model that shows that this naive (arbitrary) page placement leads to up to 30% unnecessary cache conflicts. We develop several page placement algor ...

2 UTLB: a mechanism for address translation on network interfaces

Yuqun Chen, Angelos Bilas, Stefanos N. Damianakis, Cezary Dubnicki, Kai Li

October 1998 **Proceedings of the eighth international conference on Architectural support for programming languages and operating systems**, Volume 33 , 32 Issue 11 , 5

Full text available: pdf(1.76 MB) Additional Information: full citation, abstract, references, citings, index terms

An important aspect of a high-speed network system is the ability to transfer data directly between the network interface and application buffers. Such a *direct data path* requires the network interface to "know" the virtual-to-physical address translation of a user buffer, *i.e.*, the physical memory location of the buffer. This paper presents an efficient address translation architecture, User-managed TLB (UTLB), which eliminates system calls and device interrupts from the common co ...

3 Effect of node size on the performance of cache-conscious B+-trees

Richard A. Hankins, Jignesh M. Patel

June 2003 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems**, Volume 31 Issue 1

Full text available: pdf(271.16 KB) Additional Information: full citation, abstract, references, index terms

In main-memory databases, the number of processor cache misses has a critical impact on the performance of the system. Cache-conscious indices are designed to improve performance by reducing the number of processor cache misses that are incurred during a

search operation. Conventional wisdom suggests that the index's node size should be equal to the cache line size in order to minimize the number of cache misses and improve performance. As we show in this paper, this design choice ignores additi ...

Keywords: B+-tree, cache-conscious, index

4 Improving IPC by kernel design

Jochen Liedtke

December 1993 **ACM SIGOPS Operating Systems Review , Proceedings of the fourteenth ACM symposium on Operating systems principles**, Volume 27 Issue 5

Full text available: pdf(1.39 MB) Additional Information: full citation, abstract, references, citings, index terms

Inter-process communication (ipc) has to be fast and effective, otherwise programmers will not use remote procedure calls (RPC), multithreading and multitasking adequately. Thus ipc performance is vital for modern operating systems, especially μ-kernel based ones. Surprisingly, most μ-kernels exhibit poor ipc performance, typically requiring 100 μs for a short message transfer on a modern processor, running with 50 MHz clock rate.In contrast, we achieve 5 μs; a twenty ...

5 Characterizing the caching and synchronization performance of a multiprocessor operating system

Josep Torrellas, Anoop Gupta, John Hennessy

September 1992 **ACM SIGPLAN Notices , Proceedings of the fifth international conference on Architectural support for programming languages and operating systems**, Volume 27 Issue 9

Full text available: pdf(1.52 MB) Additional Information: full citation, references, citings, index terms

6 Compiler-based I/O prefetching for out-of-core applications

Angela Demke Brown, Todd C. Mowry, Orran Krieger

May 2001 **ACM Transactions on Computer Systems (TOCS)**, Volume 19 Issue 2

Full text available: pdf(499.03 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Current operating systems offer poor performance when a numeric application's working set does not fit in main memory. As a result, programmers who wish to solve "out-of-core" problems efficiently are typically faced with the onerous task of rewriting an application to use explicit I/O operations (e.g., read/write). In this paper, we propose and evaluate a fully automatic technique which liberates the programmer from this task, provides high performance, and requires only minima ...

Keywords: compiler optimization, prefetching, virtual memory

7 Disco: running commodity operating systems on scalable multiprocessors

Edouard Bugnion, Scott Devine, Kinshuk Govil, Mendel Rosenblum

November 1997 **ACM Transactions on Computer Systems (TOCS)**, Volume 15 Issue 4

Full text available: pdf(400.76 KB) Additional Information: full citation, abstract, references, citings, index terms, review

In this article we examine the problem of extending modern operating systems to run efficiently on large-scale shared-memory multiprocessors without a large implementation effort. Our approach brings back an idea popular in the 1970s: virtual machine monitors.

We use virtual machines to run multiple commodity operating systems on a scalable multiprocessor. This solution addresses many of the challenges facing the system software for these machines. We demonstrate our approach with a prototy ...

Keywords: scalable multiprocessors, virtual machines

8 CAMERA: introducing memory concepts via visualization
Linda Null, Karishma Rao
February 2005 **ACM SIGCSE Bulletin , Proceedings of the 36th SIGCSE technical symposium on Computer science education**, Volume 37 Issue 1
Full text available: pdf(484.88 KB) Additional Information: full citation, abstract, references, index terms

CAMERA, *Cache and Memory Resource Allocation*, is a collection of workbenches for cache mapping schemes (including direct, fully associative, and set associative) and virtual memory (including paging and TLBs). Its goals are to provide users with interactive tutorials and simulations to help them better understand the fundamental concepts of memory management. Implemented in Java Swing, these workbenches allow users to observe the processes of memory to cache mapping, and virtual memory us ...

Keywords: computer memory workbenches, education, tutorial

9 Automatic pool allocation: improving performance by controlling data structure layout in the heap
Chris Lattner, Vikram Adve
May 2005 **ACM SIGPLAN Notices , Proceedings of the 2005 ACM SIGPLAN conference on Programming language design and implementation**, Volume 40 Issue 6
Full text available: pdf(215.30 KB) Additional Information: full citation, abstract, references, index terms

This paper describes *Automatic Pool Allocation*, a transformation framework that segregates distinct instances of heap-based data structures into seperate memory pools and allows heuristics to be used to partially control the internal layout of those data structures. The primary goal of this work is performance improvement, not automatic memory management, and the paper makes several new contributions. The key contribution is a new compiler algorithm for partitioning heap objects in impera ...

Keywords: cache, data layout, pool allocation, recursive data structure, static analysis

10 Application performance and flexibility on exokernel systems
M. Frans Kaashoek, Dawson R. Engler, Gregory R. Ganger, Hector M. Briceño, Russell Hunt, David Mazières, Thomas Pinckney, Robert Grimm, John Jannotti, Kenneth Mackenzie
October 1997 **ACM SIGOPS Operating Systems Review , Proceedings of the sixteenth ACM symposium on Operating systems principles**, Volume 31 Issue 5
Full text available: pdf(2.39 MB) Additional Information: full citation, references, citings, index terms

11 Automatic tiling of iterative stencil loops
Zhiyuan Li, Yonghong Song
November 2004 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 26 Issue 6
Full text available: pdf(947.69 KB) Additional Information: full citation, abstract, references, index terms

Iterative stencil loops are used in scientific programs to implement relaxation methods for numerical simulation and signal processing. Such loops iteratively modify the same array

elements over different time steps, which presents opportunities for the compiler to improve the temporal data locality through loop tiling. This article presents a compiler framework for automatic tiling of iterative stencil loops, with the objective of improving the cache performance. The article first presents a ...

Keywords: Caches, loop transformations, optimizing compilers

12 Towards a theory of cache-efficient algorithms
Sandeep Sen, Siddhartha Chatterjee, Neeraj Dumir
November 2002 **Journal of the ACM (JACM)**, Volume 49 Issue 6

Full text available: pdf(273.41 KB) Additional Information: full citation, abstract, references, index terms

We present a model that enables us to analyze the running time of an algorithm on a computer with a memory hierarchy with limited associativity, in terms of various cache parameters. Our cache model, an extension of Aggarwal and Vitter's I/O model, enables us to establish useful relationships between the cache complexity and the I/O complexity of computations. As a corollary, we obtain cache-efficient algorithms in the single-level cache model for fundamental problems like sorting, FFT, and an i ...

Keywords: Hierarchical memory, I/O complexity, lower bound

13 Binary translation and architecture convergence issues for IBM system/390
Michael Gschwind, Kemal Ebcioğlu, Erik Altman, Sumedh Sathaye
May 2000 **Proceedings of the 14th international conference on Supercomputing**

Full text available: pdf(1.44 MB) Additional Information: full citation, abstract, references, index terms

We describe the design issues in an implementation of the ESA/390 architecture based on binary translation to a very long instruction word (VLIW) processor. During binary translation, complex ESA/390 instructions are decomposed into instruction "primitives" which are then scheduled onto a wide-issue machine. The aim is to achieve high instruction level parallelism due to the increased scheduling and optimization opportunities which can be exploited by binary translation software ...

14 System-level power optimization: techniques and tools
Luca Benini, Giovanni de Micheli
April 2000 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 5 Issue 2

Full text available: pdf(385.22 KB) Additional Information: full citation, abstract, references, citings, index terms

This tutorial surveys design methods for energy-efficient system-level design. We consider electronic sytems consisting of a hardware platform and software layers. We consider the three major constituents of hardware that consume energy, namely computation, communication, and storage units, and we review methods of reducing their energy consumption. We also study models for analyzing the energy cost of software, and methods for energy-efficient software design and compilation. This survery ...

15 Implementation aspects of a SPARC V9 complete machine simulator
Bill Clarke, Adam Czezowski, Peter Strazdins
January 2002 **Australian Computer Science Communications , Proceedings of the twenty-fifth Australasian conference on Computer science - Volume 4 CRPITS '02**, Volume 24 Issue 1

Full text available: pdf(1.33 MB) Additional Information: full citation, abstract, references, index terms

In this paper we present work in progress in the development of a complete machine

simulator for the UltraSPARC, an implementation of the SPARC V9 architecture. The complexity of the UltraSPARC ISA presents many challenges in developing a reliable and yet reasonably efficient implementation of such a simulator. Our implementation includes a heavily object-oriented design for the simulator modules and infrastructure, caching of repeated computations for performance, adding an OS (system call) emu ...

Keywords: SMP, SPARC V9 ISA, UltraSPARC, complete machine simulator, execution-driven simulation, object-oriented design

16 Accelerating shared virtual memory via general-purpose network interface support

Angelos Bilas, Dongming Jiang, Jaswinder Pal Singh

February 2001 **ACM Transactions on Computer Systems (TOCS)**, Volume 19 Issue 1

Full text available: pdf(178.88 KB) Additional Information: full citation, abstract, references, index terms, review

Clusters of symmetric multiprocessors (SMPs) are important platforms for high-performance computing. With the success of hardware cache-coherent distributed shared memory (DSM), a lot of effort has also been made to support the coherent shared-address-space programming model in software on clusters. Much research has been done in fast communication on clusters and in protocols for supporting software shared memory across them. However, the performance of software virtual memory (SVM) is sti ...

Keywords: applications, clusters, shared virtual memory, system area networks

17 The Opie compiler from row-major source to Morton-ordered matrices

Steven T. Gabriel, David S. Wise

June 2004 **Proceedings of the 3rd workshop on Memory performance issues: in conjunction with the 31st international symposium on computer architecture**

Full text available: pdf(457.69 KB) Additional Information: full citation, abstract, references, index terms

The Opie Project aims to develop a compiler to transform C codes written for row-major matrix representation into equivalent codes for Morton-order matrix representation, and to apply its techniques to other languages. Accepting a possible reduction in performance we seek to compile a library of usable code to support future development of new algorithms better suited to Morton-ordered matrices.This paper reports the formalism behind the OPIE compiler for C, its status: now compiling several sta ...

Keywords: cache, paging, quadtrees, scientific computing

18 Creating and preserving locality of java applications at allocation and garbage collection times

Yefim Shuf, Manish Gupta, Hubertus Franke, Andrew Appel, Jaswinder Pal Singh

November 2002 **ACM SIGPLAN Notices , Proceedings of the 17th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 37 Issue 11

Full text available: pdf(180.20 KB) Additional Information: full citation, abstract, references, citings, index terms

The growing gap between processor and memory speeds is motivating the need for optimization strategies that improve data locality. A major challenge is to devise techniques suitable for pointer-intensive applications. This paper presents two techniques aimed at improving the memory behavior of pointer-intensive applications with dynamic memory allocation, such as those written in Java. First, we present an allocation time object

placement technique based on the recently introduced notion of *p* ...

Keywords: *JVM, Java, garbage collection, heap traversal, locality, locality based graph traversal, memory allocation, memory management, object co-allocation, object placement, prolific types, run-time systems*

19 Session I: transformation: Metrics and models for reordering transformations

Michelle Mills Strout, Paul D. Hovland

June 2004 **Proceedings of the 2004 workshop on Memory system performance**

Full text available: pdf(257.48 KB) Additional Information: full citation, abstract, references, index terms

Irregular applications frequently exhibit poor performance on contemporary computer architectures, in large part because of their inefficient use of the memory hierarchy. Run-time data, and iteration-reordering transformations have been shown to improve the locality and therefore the performance of irregular benchmarks. This paper describes models for determining which combination of run-time data- and iteration-reordering heuristics will result in the best performance for a given dataset. We pr ...

Keywords: data locality, inspector/executor, locality metrics, optimization, run-time reordering transformations, spatial locality graph, temporal locality hypergraph

20 Caching & file systems: Fast data-locality profiling of native execution

Erik Berg, Erik Hagersten

June 2005 **Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems**

Full text available: pdf(349.73 KB) Additional Information: full citation, abstract, references, index terms

Performance tools based on hardware counters can efficiently profile the cache behavior of an application and help software developers improve its cache utilization. Simulator-based tools can potentially provide more insights and flexibility and model many different cache configurations, but have the drawback of large run-time overhead.We present StatCache, a performance tool based on a statistical cache model. It has a small run-time overhead while providing much of the flexibility of simulator ...

Keywords: cache behavior, profiling tool

Results 1 - 20 of 32 Result page: **1** 2 next

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player